Goal-Directed Scientific Exploration Using Multiple Rovers

Tara Estlin, Rebecca Castaño, Ashley Davies, Darren Mutz, Gregg Rabideau, Steve Chien and **Eric Mjolsness**
Jet Propulsion Laboratory
California Institute of Technology
4800 Oak Grove Drive
Pasadena, CA 91109-8099
{firstname.lastname}@jpl.nasa.gov

Abstract

This paper describes an integrated system for coordinating multiple rover behavior with the overall goal of collecting planetary surface data. The Multi-Rover Integrated Science Understanding System combines concepts from machine learning with planning and scheduling to perform autonomous scientific exploration by cooperating rovers. The integrated system utilizes a novel machine-learning clustering component to analyze science data and direct new science activities. A distributed planning and scheduling system is employed to generate rover plans for achieving science goals, to coordinate activities among rovers, and to replan when necessary. We describe each of these components and describe how they are integrated with a planetary environment simulation.

Introduction

NASA has recently outlined a new Mars program which will have us visit the red planet over six times in the next two decades. At least four of these missions will involve rovers or other robotic craft that will be used to explore the surface of the planet and perform numerous geological, atmospheric, and other scientific experiments. In order to increase science return and enable certain types of science activities, future missions such as these will utilize large sets of rovers to gather the desired data. These rovers will need to behave in a coordinated fashion where each rover accomplishes a subset of the overall mission goals and shares any acquired information. In addition, it is desirable to have highly autonomous rovers that require little communication with scientists and engineers on Earth to perform their tasks. An autonomous rover will be able to make decisions on its own as to what exact science data should be returned and how to go about the data gathering process.

This paper discusses the Multi-Rover Integrated Science Understanding System (MISUS) (Estlin *et al.* 1999) which provides a framework for autonomously generating and achieving planetary science goals. This system integrates techniques from machine learning with planning and scheduling to enable autonomous multi-rover behavior for analyzing science data, evaluating what new science observations to perform, and deciding what steps should be taken to perform them. These techniques are also integrated with a simulation environment that can model different planetary terrains and science data within a terrain.

Science data analysis in MISUS is performed using machine-learning clustering methods, which use image and spectral mineralogical features to help classify different planetary rock types. These methods look for similarity classes of visible, rock image regions within individual spectral images and across multiple images. Output clusters are used to help evaluate scientific hypotheses and also to prioritize visible surfaces for further observation based on their "scientific interest." As the system builds a model of the rock type distribution, it continuously assembles a new set of observation goals for a team of rovers to collect from different terrain locations. Thus, the clusterer drives the science process by analyzing the current data set and then deciding what new and interesting observations should be made.

A distributed planning and scheduling component is used to determine the rover activities required to achieve requested science goals. Based on an input set of goals and each rover's initial conditions, the planner generates a sequence of activities that satisfy the goals while obeying each of the rover's resource constraints and operation rules. Furthermore, as information is acquired regarding command-execution status and actual resource utilization, the planner updates future-plan projections. Planning is distributed among the individual rovers where each rover is responsible for planning for its own activities. A central planning system is responsible for dividing up the goals among the individual rovers in a fashion that minimizes the total traversing time of all rovers.

The components described above are also integrated with a simulation environment that models multiple-rover science operations in a Mars-like terrain. Different Martian rockscapes are created for use in the simulator by using distributions over rock types, sizes and locations. When science measurements are requested from a terrain during execution, rock and mineral spectral models are used to generate sample spectra based on the type of rock being observed.

Cooperating Rovers for Science

Utilizing multiple rovers on planetary science missions has many advantages. First, multiple rovers can collect more data than a single rover. Second, multiple rovers can perform tasks that otherwise would not be possible. For instance, more complicated cooperative tasks can be accomplished, such as taking a wide baseline stereo image (which requires two cameras separated by a certain distance). Finally, multiple rovers can enhance mission success through increased system redundancy. If one rover fails, then its tasks could be quickly taken over by another rover.

Coordinating multiple distributed agents for a mission to Mars or another planet introduces some interesting new challenges for the supporting technology. Issues arise concerning communication, control and individual on-board capabilities. Many of these design decisions are related, and all of them have an impact on any on-board technologies used for the mission. For example, for an on-board science analysis system, the amount of communication bandwidth available will determine how much science data can be easily shared. This factor will also affect a planning system by determining how much each rover can coordinate with other rovers to perform tasks. The control scheme will determine which rovers execute what science gathering tasks, which affects the on-board components. For instance, some rovers may be utilized only for science data gathering, while others may be used for planning and/or science analysis. Decisions regarding the on-board capabilities of each rover can also determine the independence of a rover.

For the framework presented in this paper, we have initially chosen the configuration of a team of three rovers. Science goals are divided among the three rovers. Each rover is identical and is assumed to have a spectrometer on-board as well as other resources including a drive motor, a solar panel that provides power for rover activities, and a battery that provides backup power when solar power is not available. The battery can also be recharged using the solar panel when possible. Collected science data is immediately transmitted to the lander where it is stored in memory. The lander can only receive transmissions from one rover at a time.

Multi-Rover Science Architecture

The overall MISUS architecture is shown in Figure 1. The system is comprised of three major components:

- **Data Analysis**: A distributed machine-learning system which performs unsupervised clustering to model the distribution of rock types observed by the rovers. This system is designed to direct rover sensing to continually improve this model of the scientific content of the planetary scene.
- **Planning**: A distributed-planning system that produces rover-operation plans to achieve input rover science goals. Planning is divided between a central planner, which efficiently divides up science goals between rovers, and a distributed set of planners which each plan for operations upon an individual rover.
- **Environment simulator**: A multiple rover simulator that models different geological environments and rover-science operations within them. The simulator manages science data for each environment, tracks rover operations within the terrain, and reflects readings by rover science instruments.

MISUS operates in a closed-loop fashion where the data analysis system can be seen to take the role of the scientist driving the exploration process. Spectra data are received by individual rover clustering algorithms, which attempt to locally model the distribution of rocks according to broad classifications of rock compositions. This information is then sent to a central clusterer which integrates all gathered data into an updated global model and broadcasts the new model back to the distributed clusterers. A prioritization algorithm uses the clustering output to generate a new set of observation goals that will further improve the accuracy of the model. These goals are passed to a central planner which assigns individual rovers to goals in a fashion that will most efficiently serve the requests. Then each rover planner produces a set of actions for that rover which will achieve as many of its assigned goals as possible. These action sequences are sent to the simulator where they are executed and any gathered data is sent back to the rover clusterers. This cycle continues until enough data is gathered to produce distinct clusters for any observed rock types.

In the next few sections, we discuss each of the MISUS system components in more detail.

Data Analysis System

To perform science analysis, we use a machine-learning system which performs unsupervised clustering to model the distribution of rock types in the observed terrain. A primary feature of MISUS is that the separate rovers cooperate to form a joint consensus for the observed distribution of rock types. Through a learning process, the global distribution model keeps improving as more data is observed over time. For this demonstration prototype, the model used for this distribution is a simple K-means-like unsupervised clustering model, where each cluster represents a different rock type in the sensor space. In the present simulation, each sensor reading is a spectral measurement returning values at 14 wavelengths; learning takes place in the full 14-dimensional continuous space.

At any given time, each rover has a different location on the planetary surface and is sensing different targets. So each rover has its own distinct segment of the overall dataset, stored locally in its data buffer. Over time, each rover collects a new set of data points, or 14-dimensional spectrum readings, adding it to its existing store of data points. Clustering is initiated after each rover has obtained new observations. A sample

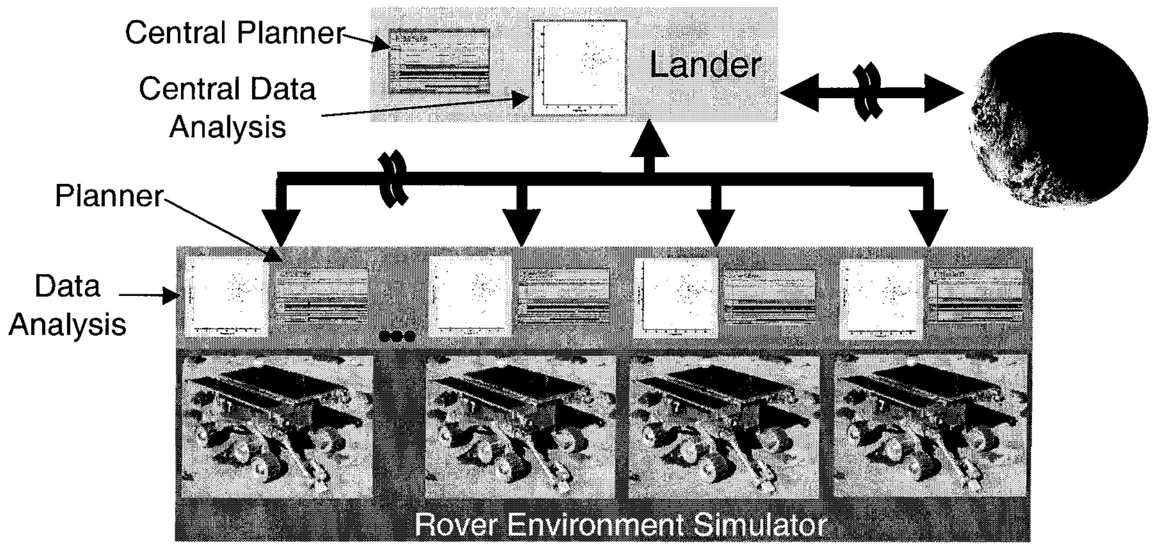

Figure 1: MISUS Architecture Diagram

cluster model (shown for 2 of 14 dimensions) is shown in Figure 2.

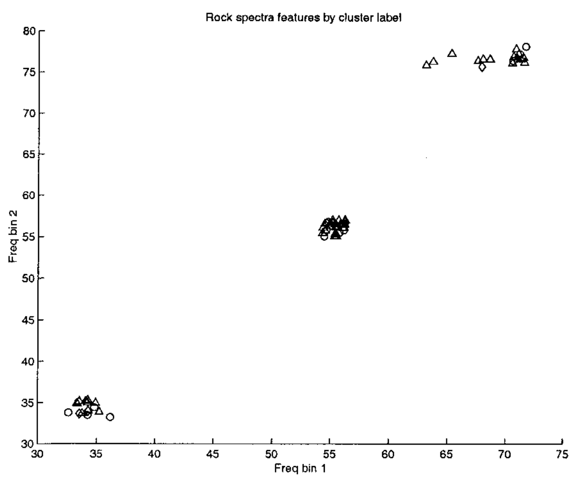

Figure 2: Example spectra feature space

Clustering is based on the EM (Expectation-Maximization) algorithm, an iterative optimization procedure, which normally requires several passes over the entire data. Rovers must share information through a power-expensive communication channel. Thus, rather than send its local dataset to one or more other rovers, the distributed clustering algorithm allows a rover to send only a small set of parameters which summarizes its local data. Each rover's model parameters are computed locally, then sent to a central clusterer which integrates them into an updated global model (which is also a small set of parameters) and broadcasts that model to all rovers in the system. Each rover takes this global model into account when making its local estimate. This process continues iteratively until convergence. This scheme trades off some accuracy in the global model in order to minimize communication. In the limit of large datasets, this scheme approximates the equivalent non-distributed clustering model (where one processor may examine all the data at once) more and more closely. In particular, the distributed version of the clustering model follows a development similar to that in (Tsioutsias & Mjolsness 1994) for partitioned neural networks.

The clustering model in this initial prototype system may be viewed as the scientific end-product of the exploration. The overall purpose of the system is to increase the accuracy of the clustering model by obtaining sensor readings in regions that are likely to improve the model. An update of the clustering model determines new planetary locations to be explored by the rovers. These locations are sent as formal goals by the learner to the planner.

A very simple heuristic for goal selection is used in the current system. A constant number G of new spatial targets will be specified for each cluster. For each cluster, two of the G spatial targets are chosen by first finding the two mutually most distant points (in physical space) of that rock type, then selecting a point in space stochastically from within a neighborhood of each of those 2 points. These goals are given high priority. The rest of the G targets are chosen from neighborhoods of randomly selected rocks in the cluster, and are given lower priority. The idea of this heuristic is to bias the system toward exploration in extremal directions, as well as to explore the rock distribution in a way which balances effort equally between rock types (thus avoiding, say, spending undue energy on a very common rock type at the expense of rare rock types).

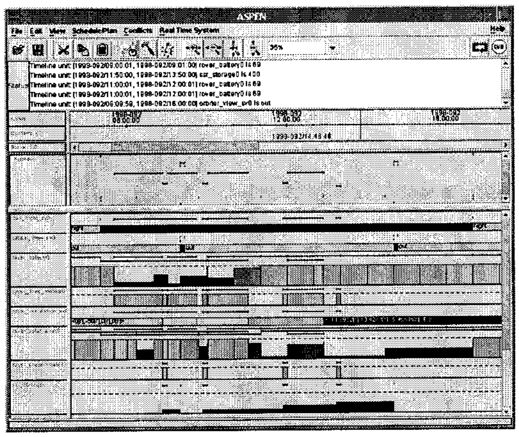

Figure 3: Example CASPER plan

Planning System

To produce individual rover plans, we used a distributed version of the CASPER (Continuous Activity Scheduling, Planning, Execution and Replanning) system (Chien *et al.* 2000). CASPER is a dynamic-planning application framework that can be tailored to specific domains. For this application, CASPER inputs a set of science goals and produces the necessary rover-activity sequence. This sequence is generated by utilizing an *iterative repair* algorithm (Zweben *et al.* 1994), which classifies conflicts and attacks them each individually. Conflicts occur when a plan constraint has been violated where this constraint could be temporal or involve a resource, state or activity parameter. Conflicts are resolved by performing one or more schedule modifications such as moving, adding, or deleting activities.

A rover that is at the incorrect location for a scheduled science activity is one type of conflict. Resolving this particular conflict involves adding a traverse command to send the rover to the designated site. Other conflicts may include having more than one rover communicating with the lander at a time or having too many activities scheduled for one rover, which oversubscribed its power resources. The iterative repair algorithm continues until no conflicts remain in the schedule, or a timeout has expired. Figure 3 shows an example rover-plan displayed in the CASPER GUI.

To support missions with multiple rovers, we developed a distributed version of CASPER where it is assumed each rover has an on-board planner. There is also a central planner, which could be located on a lander or on one of the rovers. This distributed approach allows rovers to plan for themselves and/or for other rovers. The central planner develops an abstract plan for all rovers, while each rover planner develops a detailed, executable plan for its own activities. The central planner also acts as a router, taking a global set of goals and dividing it up among the rovers. For example, a science goal may request an image of a particular rock without concern for which rover acquires the image. The central planner could assign this goal to the rover that is closest to the rock in order to minimize the traversals of all rovers.

To achieve a high-level of responsiveness for each onboard rover planner, we utilize a continuous planning approach. Each rover planner has a current goal set, a current state, a current plan and state projections in the future for that plan. At any time, an incremental update to the goals or current state may change the current plan. This update may be an unexpected event or simply time progressing forward. Each rover planner is then responsible for maintaining a plan consistent with the most current information. The current plan is the planner's estimation as to what it expects to happen in the world if things go as expected. However, since things rarely go exactly as expected, each planner stands ready to continually modify the plan. Iterative repair techniques, as mentioned previously, enable incremental changes to the goals, initial state or plan, and then iteratively resolve any conflicts that may arise.

One of the dominating characteristics of the multi-rover application is rover traversals to designated waypoints. Decisions must be made not only to satisfy the requested goals, but also to provide more optimal schedules. CASPER can consider optimization goals during the repair process. As certain types of conflicts are resolved, heuristics are used to guide the search towards making decisions that will produce higher quality schedules. For this application, we have implemented heuristics based on techniques from the Multi-Traveling Salesmen Problem (MTSP), which is similar to the Traveling Salesman Problem (TSP). For MTSP, at least one member of a sales team must visit each city such that total traveling time is minimized. Both the central and rover planners utilize the MTSP heuristics. In previously reported results, they were shown to make a significant impact in reducing traversal distance and expected execution time (Rabideau, Estlin, & Chien 1999).

Environment Simulator

The environment simulator is designed to provide a source of data for the science analysis system by simulating the science gathering activities of the rover. Given the current science scenario, this entails the generation of an environment and the simulation of rover data gathering activities within the environment.

Generation of the environment requires producing a field of rocks for the rovers to traverse. The rock field is generated as a plane with rocks of various sizes embedded at various depths. The simulator maintains information about the mineral composition of each rock, and the spectrum that would correspond to its mineral composition. The size and spatial distributions of the rockfield were developed by examining distributions of rocks observed by the Viking Landers, Mars Lander and Mars Pathfinder. The distribution of minerals that can occur in rocks was developed in collaboration with planetary geologists at JPL, and the spectra associated with rocks are generated from the spectra of the component minerals via a linear-mixing model.

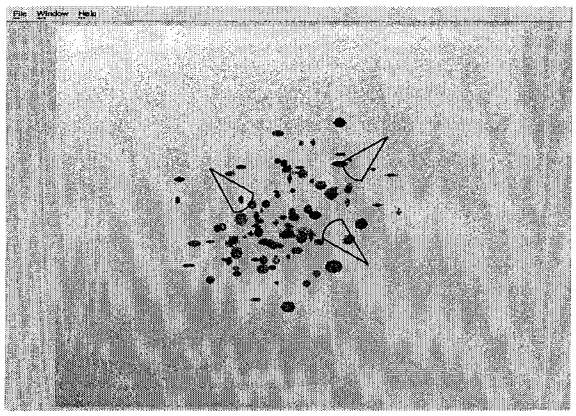

Figure 4: Overhead view of simulated rockscape. Wedges denote different rovers' spectrometers' fields of view.

The simulation of the rover activities was done at a coarse level. Such considerations as kinematics and obstacle avoidance were not modeled. Other considerations, such as power consumption and memory management were only modeled by the planner for plan generation. The rovers were essentially modeled as roving spectrometers by the simulator. Figure 4 shows several rovers and their spectrometer reaches modeled in a sample rockscape. The simulation of rover activities was accomplished by executing the plan generated by the planner, consisting of a list of movement, rotation, and instrument commands. The simulator would then, from the location and direction specified by the movement and rotation commands, determine whether or not a rock was visible by the boresighted spectrometer. If so, the simulator would perturb the spectra in an amount proportional to the distance of the rover from the rock in order to simulate instrument noise, and store the spectrum for later communication to the relevant clusterer. After all of the activities in a plan were executed by the simulator (i.e. moves, turns, and data gathering activities), the data was communicated to each clusterer via synchronization agents. The simulator would then wait for the next plan.

Related Work

The idea of having a scientific discovery system direct future experiments is present in a number of other systems (Nordhausen & Langley 1993; Rajamoney 1990), however none of these systems examine the problem of planetary science, interact with an environment simulator or are integrated with a planning system that can create a command sequence to perform the necessary experiments.

There has also been a significant amount of work on cooperating robots, in the form of distributing planning approaches (Brummit & Stentz 1988) and behavioral approaches (Mataric 1995; Parker 1999). However most of these systems to not reason about high-level mission goals and none of these systems utilize a learning component to drive science experiments.

Conclusions

This paper outlines a framework for coordinating multiple rover behavior in generating and achieving geological science goals. This system integrates techniques from machine learning and planning and scheduling to autonomously analyze and request new science data and generate the action sequences to retrieve that data. We discuss a number of integration issues including developing shared goal and plan representations, coordinating systems asynchronously, and adjusting interface parameters to best serve the overall system goal. We hope the techniques and issues presented in this paper will prove useful to other designers of integrated systems.

Acknowledgments

This work was performed ~~by~~ at the Jet Propulsion Laboratory, California Institute of Technology, under contract with the National Aeronautics and Space Administration. The authors acknowledge the invaluable contributions of Alex Gray and Tobias Mann for their help in defining an early version of the data analysis system and environmental simulator.

References

Brummit, B., and Stentz, A. 1988. GRAMMPS: A generalized mission planner for multiple mobile robots in unstructured environments. In *Proceedings of the IEEE Conference on Robots and Automation.*

Chien, S.; Knight, R.; Stechert, A.; Sherwood, R.; and Rabideau, G. 2000. Using iterative repair to improve responsiveness of planning and scheduling. In *Proceedings of the Fifth International Conference on Artificial Intelligence Planning and Scheduling.*

Estlin, T.; Gray, A.; Mann, T.; Rabideau, G.; Castano, R.; Chien, S.; and Mjolsness, E. 1999. An integrated system for multi-rover scientific exploration. In *Proceedings of the Sixteenth National Conference on Ariticial Intelligence.*

Mataric, M. 1995. Designing and understanding adaptive group behavior. *Adaptive Behavior* 4(1):51–80.

Nordhausen, B., and Langley, P. 1993. An integrated framework for empirical discovery. *Machine Learning* 12:17–47.

Parker, L. E. 1999. Cooperative robotics for multi-target observation. *Intelligent Automation and Soft Computing* 5(1):5–19.

Rabideau, G.; Estlin, T.; and Chien, S. 1999. Working together: Automatic generation of command sequences for multiple cooperating rovers. In *Proceedings of the 1999 IEEE Aerospace Conference.*

Rajamoney, S. 1990. A computational approach to theory revision. In Shrager, J., and Langley, P., eds., *Computational Models of Scientific Discovery and Theory Formation.* San Mateo, CA: Morgan Kaufman. 225–254.

Tsioutsias, D., and Mjolsness, E. 1994. Optimization dynamics for partitioned neural networks. *International Journal of Neural Systems* 5(4).

Zweben, M.; Daun, B.; Davis, E.; and Deale, M. 1994. Scheduling and rescheduling with iterative repair. In Zweben, M., and Fox, M., eds., *Intelligent Scheduling.* San Francisco, CA: Morgan Kaufmann. 241–256.